Context-Dependent Manifold Learning in Dynamical Systems: A Neuromodulated Constrained Autoencoder Approach

Jérôme Adriaens JADRIAENS@ULIEGE.BE
Guillaume Drion GDRION@ULIEGE.BE
Pierre Sacré P.SACRE@ULIEGE.BE
Neuroengineering Lab, Department of Electrical Engineering and Computer Science, University of Liège, Allée de la Découverte 11

Editors: List of editors' names

Abstract

This work introduces a novel approach to context-dependent manifold learning in dynamical systems using a modulated constrained autoencoder (cAE). Classic dimensionality reduction methods often fail to account for context-dependent relationships in data without explicitly reducing the context combined with the original input. However, these relationships are critical when physical parameters or environmental conditions vary. Building on the constrained autoencoder framework, which imposes geometric constraints to ensure smooth manifold representations and proper projections, we incorporate neuromodulation to enable context-dependent learning. Neuromodulation is a fundamental mechanism that uses neuromodulators to tune neuronal properties and circuit function dynamically. It is essential for generating flexible brain states and complex behaviors. Our method effectively integrates contextual information into the constrained autoencoder framework, allowing for context-dependent dimensionality reduction. This advancement has significant implications for learning smooth, context-aware manifolds in dynamical systems.

Keywords: Nonlinear dimensionality reduction, Neuromodulation, Context-dependent learning, Constrained autoencoder

1. Introduction & related work

Dimensionality reduction is fundamental for analysing high-dimensional data. While linear methods like PCA are widely used, they often fail to capture complex nonlinear relationships. This limitation led to the development of nonlinear approaches such as autoencoders (Wang et al., 2016) and kernel PCA (Schölkopf et al., 1997). However, traditional linear and nonlinear dimensionality reduction methods commonly overlook context, which critically influences data relationships, especially in dynamical systems with varying physical parameters or environmental conditions. The traditional To address this without augmenting input dimensions, we seek a mechanism that subtly adapts the projection itself.

To bridge this gap, we propose a novel approach to context-dependent dimensionality reduction, leveraging the cAE framework (Otto et al., 2023). This framework, which utilizes Riemannian optimization on the biorthogonal manifold, ensures smooth, data-respecting manifold representations—a key advantage over unstructured linear or nonlinear methods. To enable adaptive learning, we integrate neuromodulation that dynamically tunes the cAE activation functions with a context vector, inspired by biological mechanisms (Vecoven et al., 2020). Our network adapts its dimensionality reduction to external signals or changing physical parameters. This integration is further enhanced by leveraging geometric optimization for manifold learning (Friedl et al., 2025), aligning well with the cAE geometrically constrained output.

This work initially explores neuromodulation integration within the cAE framework. We demonstrate its effectiveness for context-dependent dimensionality reduction on a 16-degree-of-freedom (DoF) pendulum dynamical system with varying context parameters.

2. Neuromodulated Constrained Autoencoder

This section details our neuromodulated cAE, first introducing its framework for well-structured manifold learning, then explaining neuromodulation role in context-dependent adaptation. See Figure 1 for a visual overview.

2.1. Constrained Autoencoders

The cAE, as introduced by (Otto et al., 2023), is designed to learn smooth, embedded submanifolds of the data space. It achieves this by enforcing a critical geometric property: the composition of its encoder and decoder functions forms an idempotent projection operator. Mathematically, this means the cAE encoder ρ and decoder φ satisfy $\rho \circ \varphi = \mathrm{Id}$, making $P = \varphi \circ \rho$ a smooth, idempotent mapping. This ensures the learned manifold $\hat{\mathcal{M}} = \mathrm{Range}(P)$ is indeed a smooth embedded submanifold of the data space (Michor, 2008). To enforce this property in a neural network, the encoder $\rho = \rho^{(1)} \circ \cdots \circ \rho^{(L)}$ and the decoder $\varphi = \varphi^{(L)} \circ \cdots \circ \varphi^{(1)}$ are built pairwise from layer pairs (l). Each layer uses biorthogonal weight matrices $\boldsymbol{\Psi}_l$ and $\boldsymbol{\Phi}_l$, and smooth activation function pairs (σ_-, σ_+) which are the inverse of one another:

$$\rho^{(l)}(\boldsymbol{x}^{(l)}) = \sigma_- \left(\boldsymbol{\Psi}_l^T (\boldsymbol{x}^{(l)} - \boldsymbol{b}_l) \right), \quad \varphi^{(l)}(\boldsymbol{z}^{(l-1)}) = \boldsymbol{\Phi}_l \sigma_+ \left(\boldsymbol{z}^{(l-1)} \right) + \boldsymbol{b}_l \tag{1}$$

This construction inherently includes the constraint $\boldsymbol{\Psi}_l^T \boldsymbol{\Phi}_l = \boldsymbol{I}$. The biorthogonality constraint is enforced during training either by projection methods or, as in Friedl et al. (2025) and adopted in our work, through Riemannian optimization on the biorthogonal manifold using the `geoopt` library. More details about the cAE can be found in Appendix A.

2.2. Neuromodulation

In addition to the geometric constraints imposed by the cAE, we introduce neuromodulation in the framework to enable context-dependent learning. In biological systems, neuromodulation is a crucial process that grants the nervous system its remarkable flexibility, dynamically tuning the fundamental properties of neurons and networks(Bargmann and Marder, 2013). This dynamic regulation is vital for generating flexible brain states and complex behaviors. From a machine learning perspective, neuromodulation allows a network to flexibly adjust its internal representations in response to varying environments or tasks (Vecoven et al., 2020).

In this work, we achieve context-dependent learning by replacing the fixed activation functions from Section 2.1 with parameterized versions, $\sigma_\pm(\cdot; \boldsymbol{\alpha}^{(l)})$. These layer-specific parameters, $\boldsymbol{\alpha}^{(l)}$, are generated through a two-step process driven by a context vector $\boldsymbol{c}$. Initially, the context vector $\boldsymbol{c}$ is passed through a fully connected layer to produce a global neuromodulation signal $\boldsymbol{s}$. Subsequently, for each layer l, this signal $\boldsymbol{s}$ is multiplied by a layer-pair specific weight matrix, $\boldsymbol{W}_l^T$. This mechanism allows the activation functions to adjust their nonlinear transformations flexibly in response to the given context. Figure 1 illustrates how the context vector $\boldsymbol{c}$ influences these activation functions. More details can be found in Appendix B.

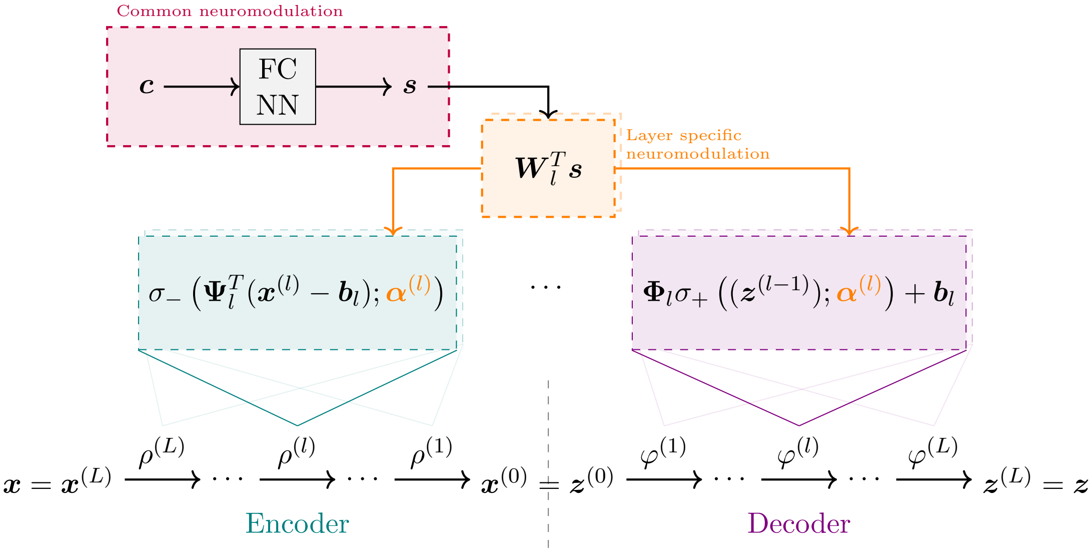
Common neuromodulation
$\boldsymbol{c}$
FC NN
$\boldsymbol{s}$
$\boldsymbol{W}_l^T \boldsymbol{s}$
Layer specific neuromodulation
$\sigma_- \left(\boldsymbol{\Psi}_l^T (\boldsymbol{x}^{(l)} - \boldsymbol{b}_l); \boldsymbol{\alpha}^{(l)}\right)$
$\cdots$
$\boldsymbol{\Phi}_l \sigma_+ \left((\boldsymbol{z}^{(l-1)}); \boldsymbol{\alpha}^{(l)}\right) + \boldsymbol{b}_l$
$\boldsymbol{x} = \boldsymbol{x}^{(L)} \xrightarrow{\rho^{(L)}} \cdots \xrightarrow{\rho^{(l)}} \cdots \xrightarrow{\rho^{(1)}} \boldsymbol{x}^{(0)} = \boldsymbol{z}^{(0)} \xrightarrow{\varphi^{(1)}} \cdots \xrightarrow{\varphi^{(l)}} \cdots \xrightarrow{\varphi^{(L)}} \boldsymbol{z}^{(L)} = \boldsymbol{z}$
Encoder
Decoder

Figure 1: Scheme of the inclusion of neuromodulation in the cAE. The context vector $\boldsymbol{c}$ generates a neuromodulation signal $\boldsymbol{s}$. This signal is then multiplied by a layer pair weight matrix $\boldsymbol{W}_l$ to obtain the activation function parameters $\boldsymbol{\alpha}^{(l)}$.

3. Experiments & Results

The experiments consist in two parts, both based on a 16-DoF pendulum. The first, referred to as the *standard pendulum experiment*, modifies the setup from Friedl et al. (2025) by sampling randomly the lengths of the first four links uniformly in the range $[0.35, 0.65]$m, while keeping the coupling of the last 12 DoF unchanged. For the second, the *context-dependent pendulum experiment*, we adapt the pendulum system to integrate context-dependent context-dependent coupling, where the coupling function links the lengths of the four DoF and their joint angles. It is important to note that the second task is significantly more complex than the first one as can be seen in Figure 2. The exact coupling and data generation process are provided in Appendix C. For both experiments, we compare three architectures to assess the impact of neuromodulation and contextual information:

1. *cAE*: A constrained autoencoder without modulation.

2. *IcAE*: A modulated constrained autoencoder with an irrelevant context ($\boldsymbol{c} = \mathbf{0}$), serving as a control for increased model capacity.

3. *McAE*: A modulated constrained autoencoder with a relevant context.

This evaluation aims to demonstrate that incorporating relevant contextual information significantly improves dimensionality reduction. We report the root mean square error (RMSE) distributions for position and first-derivative (velocity) reconstruction on an unseen test set in Figure 2. In the standard pendulum experiment, all three architectures yield comparable performance, aligning with theoretical expectations given the context-independent coupling. However, the context-dependent pendulum experiment reveals compelling insights: while the IcAE and cAE exhibit similar reconstruction errors, the McAE demonstrates significantly superior performance, reducing both position and velocity reconstruction errors by a factor of 2 to 3. This observed difference in performance empirically supports that the McAE effectively leverages contextual information, validating our neuromodulation approach for addressing context-dependent learning challenges.

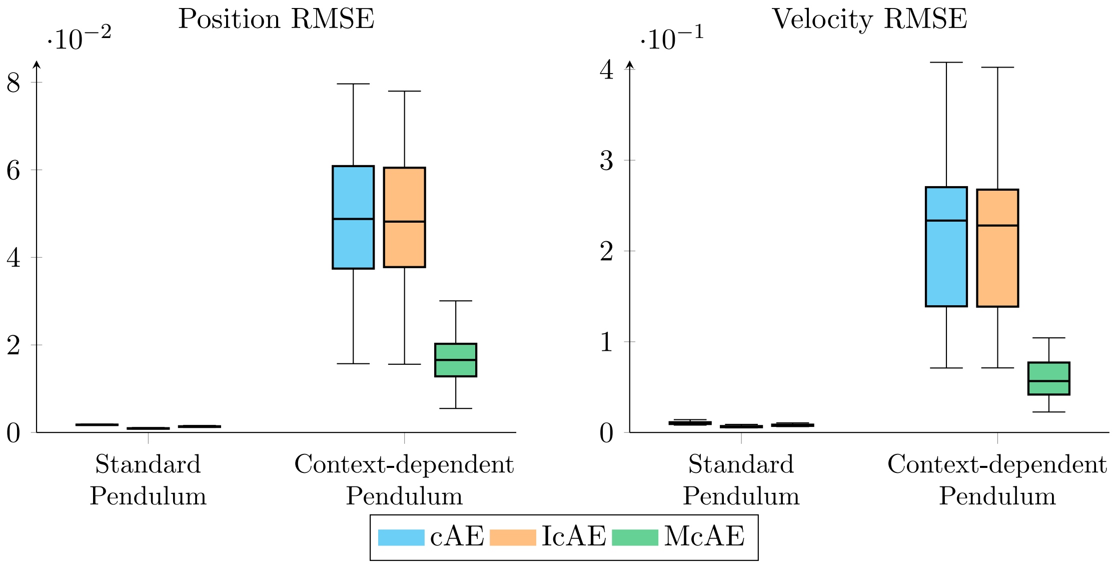

Figure 2: Distribution of the root mean square error (RMSE) using the test set on the three architectures. The RMSE is computed on 1 test trajectory of 3001 states with the same link lengths. *Left:* RMSE of the reconstruction of the pendulum DoFs. *Right:* RMSE of the first-order derivative reconstruction of the pendulum DoFs.

4. Conclusion & Perspectives

This work introduces a novel approach for context-dependent manifold learning using neuromodulated cAE. Preliminary results demonstrate the potential of our method, particularly in capturing context-dependent relationships in dynamical systems. This work is preliminary, and future efforts will focus on refining and rigorously validating our approach. This includes a deeper inspection of the latent space under various context changes, and further refinements to the neuromodulation mechanism itself. Additionally, we aim to extend its applicability to higher-dimensional and more dynamical systems, such as the three-state vortex model from Otto et al. (2023) and the 192 DoF rope simulation described in Friedl et al. (2025), to fully validate its generalizability. We believe this method has the potential to significantly advance context-dependent manifold learning and enable new applications in dynamical systems.

References

Cornelia I. Bargmann and Eve Marder. From the connectome to brain function. *Nature Methods*, 10(6):483–490, June 2013. ISSN 1548-7105. doi: 10.1038/nmeth.2451.

Katharina Friedl, Noémie Jaquier, Jens Lundell, Tamim Asfour, and Danica Kragic. A riemannian framework for learning reduced-order lagrangian dynamics. In *Intl. Conf. on Learning Representations (ICLR)*, 2025.

Klaus Glashoff and Michael M. Bronstein. Optimization on the biorthogonal manifold, September 2016.

Max Kochurov, Rasul Karimov, and Serge Kozlukov. Geoopt: Riemannian optimization in pytorch, 2020.

Peter Michor. *Topics in Differential Geometry*, volume 93 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, Rhode Island, July 2008. ISBN 978-0-8218-2003-2 978-1-4704-1161-9. doi: 10.1090/gsm/093.

Samuel E. Otto, Gregory R. Macchio, and Clarence W. Rowley. Learning Nonlinear Projections for Reduced-Order Modeling of Dynamical Systems using Constrained Autoencoders, September 2023.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In Wulfram Gerstner, Alain Germond, Martin Hasler, and Jean-Daniel Nicoud, editors, *Artificial Neural Networks — ICANN'97*, pages 583–588, Berlin, Heidelberg, 1997. Springer. ISBN 978-3-540-69620-9. doi: 10.1007/BFb0020217.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Nicolas Vecoven, Damien Ernst, Antoine Wehenkel, and Guillaume Drion. Introducing neuromodulation in deep neural networks to learn adaptive behaviours. *PLOS ONE*, 15 (1):e0227922, January 2020. ISSN 1932-6203. doi: 10.1371/journal.pone.0227922.

Yasi Wang, Hongxun Yao, and Sicheng Zhao. Auto-encoder based dimensionality reduction. *Neurocomputing*, 184:232–242, April 2016. ISSN 0925-2312. doi: 10.1016/j.neucom.2015.08.104.

Acknowledgments

This work was supported by the Belgian Government through the FPS Policy and Support (BOSA) grant NEMODEI.

Appendix A. Constrained autoencoder details

The constrained autoencoder was initially proposed by Otto et al. (2023). They impose that their autoencoder is a projection to leverage the autoencoder for projection-based reduced order modeling. The autoencoder is composed of an encoder ρ and a decoder φ such that the encoder maps the data space $\mathcal{X}$ to a latent space $\mathcal{Z}$, and the decoder reconstructs the data from the latent space. By imposing $\rho \circ \varphi = \mathrm{Id}_{\mathcal{Z}}$, we ensure that $P = \varphi \circ \rho$ is a projection. In this case, P is smooth and idempotent, so $\hat{\mathcal{M}} = \mathrm{Range}(P)$ is a smooth embedded submanifold of $\mathcal{X}$ (Otto et al., 2023; Michor, 2008). The encoder ρ and decoder φ are defined as a composition of layers such that $\rho = \rho^{(1)} \circ \cdots \circ \rho^{(L)}$ and $\varphi = \varphi^{(L)} \circ \cdots \circ \varphi^{(1)}$, where $\rho^{(l)} : \mathcal{R}^{n_l} \to \mathcal{R}^{n_{l-1}}$ and $\varphi^{(l)} : \mathcal{R}^{n_{l-1}} \to \mathcal{R}^{n_l}$. Denoting d the latent space dimension and n the input space dimension, $d = n_0 \leq \cdots \leq n_L = n$.

A.1. Biorthogonal layers

To ensure the geometric properties of the cAE, the encoder ρ and decoder φ are constructed with specific layer structures. Each layer transformation for the encoder and decoder can be expressed as:

$$\rho^{(l)}(\boldsymbol{x}^{(l)}) = \sigma_- \left(\boldsymbol{\Psi}_l^T (\boldsymbol{x}^{(l)} - \boldsymbol{b}_l) \right) \quad \text{and} \quad \varphi^{(l)}(\boldsymbol{z}^{(l-1)}) = \boldsymbol{\Phi}_l \sigma_+ \left(\boldsymbol{z}^{(l-1)} \right) + \boldsymbol{b}_l, \tag{2}$$

where $\boldsymbol{\Psi}_l \in \mathcal{R}^{n_{l-1} \times n_l}$ and $\boldsymbol{\Phi}_l \in \mathcal{R}^{n_l \times n_{l-1}}$ are the weight matrices for the encoder and decoder layers, respectively. The pair (σ_-, σ_+) consists of smooth activation functions that are inverses of one another, and $\boldsymbol{b}_l$ are the bias vectors.

For each layer to satisfy the idempotent property $\rho^{(l)} \circ \varphi^{(l)} = \mathrm{Id}_{\mathcal{R}^{n_{l-1}}}$, a crucial condition must be met by the weight matrices:

$$\boldsymbol{\Psi}_l^T \boldsymbol{\Phi}_l = \boldsymbol{I}_{\mathcal{R}^{n_{l-1}}}. \tag{3}$$

This defines $\boldsymbol{\Phi}_l$ and $\boldsymbol{\Psi}_l$ as biorthogonal matrices (Glashoff and Bronstein, 2016). This condition can be enforced in several ways. The approach from Otto et al. (2023) involves an overparametrization of weight matrices followed by a projection map onto the biorthogonal manifold, though this introduces additional loss terms. Building upon this, Friedl et al. (2025) proposed an alternative: they ensure biorthogonality by using a specific geometric optimization framework, minimizing their loss function via Riemannian optimization directly on the biorthogonal manifold. Following their methodology, we adopt this geometric optimization approach for our training process, implemented using the Python package `geoopt` (Kochurov et al., 2020).

A.2. Activation functions

The activation functions used in our cAE are a crucial component, designed to be smooth and invertible, with the encoder and decoder utilizing inverse pairs (σ_-, σ_+). These specific functions were first presented by Otto et al. (2023) as part of the cAE framework. They are derived from a particular form of hyperbola where the upper and lower branches are reflections of one another with respect to the axis $y = x$.

More formally, the expression for this inverse pair is:

$$\sigma_{\pm}(\boldsymbol{x}) = \frac{b\boldsymbol{x}}{a} \mp \frac{\sqrt{2}}{a\sin(\alpha)} \pm \frac{1}{a}\sqrt{\left(\frac{2\boldsymbol{x}}{\sin(\alpha)\cos(\alpha)} \mp \frac{\sqrt{2}}{\cos(\alpha)}\right)^2 + 2a},$$

where

$$a = \csc^2(\alpha) - \sec^2(\alpha), \qquad b = \csc^2(\alpha) + \sec^2(\alpha), \qquad \text{with } 0 < \alpha < \pi/4.$$

Notably, the parameter α in these equations is the target for our neuromodulation, allowing the activation functions to adapt based on context (details in Appendix B). A visualization of these activation functions for different values of α is provided in Figure 3.

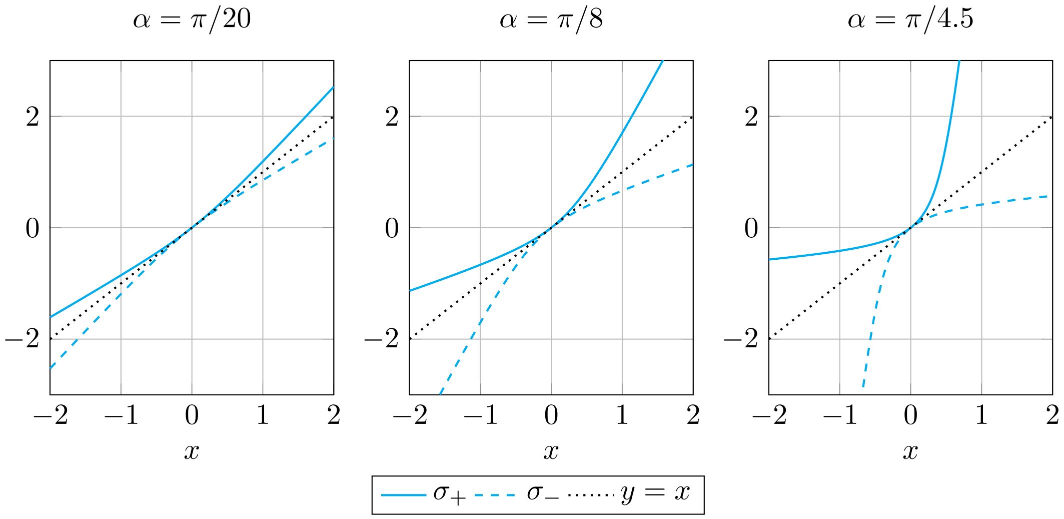

Figure 3: Activation functions $\sigma_{\pm}$ for different values of α.

Appendix B. Neuromodulation details

Neuromodulation in biological neural systems refers to the process by which neuromodulators (such as dopamine or serotonin) modulate neuronal activity and synaptic properties, enabling adaptive responses. Unlike classical neurotransmission, neuromodulation can act broadly across neural circuits, allowing the same network to produce different outputs depending on the neuromodulatory context (Bargmann and Marder, 2013). In artificial neural networks, this concept translates into dynamically adjusting network parameters based on contextual information, allowing the same network architecture to exhibit different behaviors depending on the input context. In our neuromodulated constrained autoencoder, we implement context-dependent activation functions where the activation parameters $\boldsymbol{\alpha}^{(l)}$ for a pair of layers l are modulated by a context vector $\boldsymbol{c}$.

Modulated activation functions The encoder and decoder layers in our method use context-dependent activation functions:

$$\rho^{(l)}(\boldsymbol{x}^{(l)}) = \sigma_{-}\left(\boldsymbol{\Psi}_l^T(\boldsymbol{x}^{(l)} - \boldsymbol{b}_l); \boldsymbol{\alpha}^{(l)}\right), \tag{4}$$

$$\varphi^{(l)}(\boldsymbol{z}^{(l-1)}) = \boldsymbol{\Phi}_l \sigma_{+}\left(\boldsymbol{z}^{(l-1)}; \boldsymbol{\alpha}^{(l)}\right) + \boldsymbol{b}_l, \tag{5}$$

where the parameter α from Section A.2 becomes variable and is specific to each input dimension, such that it writes now $\boldsymbol{\alpha}^{(l)} \in \mathcal{R}^{n_{l-1}}$. In addition, the values $\boldsymbol{\alpha}^{(l)}$ are constrained to lie within a specific range $[10^{-5}, \frac{\pi}{8}]$ which is more restrictive than the initial setting with the range $]0, \frac{\pi}{4}[$. The reason behind this can be shown in Figure 4 where we can see that for α values close to $\pi/4$, we have very steep slopes which can cause issues with training.

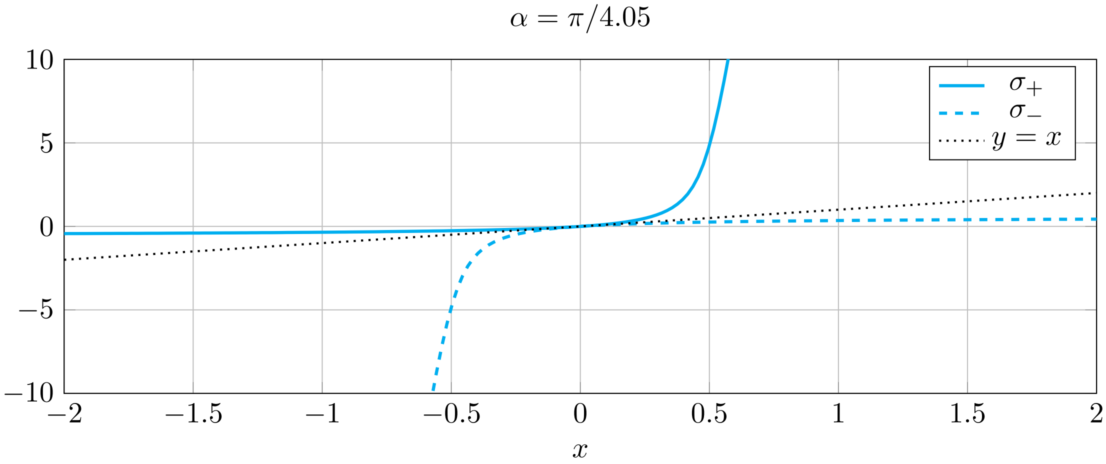

Figure 4: Activation functions $\sigma_\pm$ for α near $\frac{\pi}{4}$.

The neuromodulation process follows a two-stage mechanism:

Stage 1: Context processing The context vector $\boldsymbol{c} \in \mathcal{R}^{d_c}$ is first processed through a fully-connected neural network to generate a neuromodulation signal:

$$\boldsymbol{s} = f_{\text{nmd}}(\boldsymbol{c}; \boldsymbol{\theta}), \tag{6}$$

where f_{nmd} is a multilayer perceptron with parameters $\boldsymbol{\theta}$, and $\boldsymbol{s} \in \mathcal{R}^{d_s}$ is the resulting neuromodulation signal. We denote d_s and d_c as the dimensions of the neuromodulation signal and context vector, respectively.

Stage 2: Layer-specific modulation For each pair of layers l, the neuromodulation signal is transformed into intermediate layer-pair-specific activation parameters:

$$\bar{\boldsymbol{\alpha}}^{(l)} = \boldsymbol{W}_l^T \boldsymbol{s}, \tag{7}$$

where $\boldsymbol{W}_l \in \mathcal{R}^{d_s \times n_{l-1}}$ is a learnable transformation matrix specific to layer pair l. Then, to ensure that the activation parameters $\boldsymbol{\alpha}^{(l)}$ lie within the desired range, we apply a scaled and shifted sigmoid function:

$$\boldsymbol{\alpha}^{(l)} = (\alpha_{\max} - \alpha_{\min}) \cdot \textsc{sigmoid}\left(\bar{\boldsymbol{\alpha}}^{(l)}\right) + \alpha_{\min}, \tag{8}$$

$$\textsc{sigmoid}(\boldsymbol{x}) = \frac{1}{1 + \exp\left(-\boldsymbol{x}\right)} \tag{9}$$

where $\alpha_{\min}$ and $\alpha_{\max}$ are the minimum and maximum values for the activation parameters, respectively, and $\boldsymbol{\alpha}^{(l)} \in \mathcal{R}^{n_{l-1}}$ contains the parameters for the activation functions in that layer pair.

Appendix C. Experimentations Details

The experimental setup consists of two main parts, both centered on a 16-DoF pendulum system. The first four pendulum links are modeled in *MuJoCo* (Todorov et al., 2012) as a capsule with a fixed radius of 0.05 m and a mass of 1.0 kg, while the other DoF come from the coupling with the first 4 joints. The context vector $\boldsymbol{c}$ consists of the four link lengths (l_1 to l_4).

C.1. Standard coupling

In the first experiment, the initial 4 DoF (q_1 to q_4) correspond to the angles of the first four pendulum links. For each simulation, the lengths of these four links (l_1 to l_4) are randomly sampled from a uniform distribution in the range $[0.35, 0.65]$ m. The remaining 12 DoF (q_5 to q_{16}) are generated using nonlinear coupling functions that depend only on the first four angles q_1 to q_4, as detailed in Table 1(*a*). This setup allows us to investigate how the constrained autoencoder captures the underlying manifold structure when the context (link lengths l_i) varies but does not directly affect the coupling. In this setup, we want to ensure that the potential differences in performance are only due to the fact that the modulated networks have more parameters (since they can learn parameters for the activation functions), but we expect to have similar results.

C.2. Context-dependent coupling

The second experiment extends the first by introducing context-dependent coupling. Here, the nonlinear coupling functions for the last 12 DoF are modified to explicitly depend on both the angles (q_1 to q_4) and the corresponding link lengths (l_1 to l_4), as shown in Table 1(*b*). This creates a scenario where the context vector (link lengths) directly influences the relationships between the DoF, making the underlying manifold structure context-dependent. This experiment aims to show that relevant context information is crucial to learn the manifold structure effectively and that the modulated constrained autoencoder can use this information to improve its performance.

C.3. Data collection

This section details the procedure for collecting data for our dataset. We begin by uniformly sampling the first four link lengths from the range $[0.35, 0.65]$m. Each such pendulum configuration is then simulated in *MuJoCo* for 3s with a timestep of 1ms. For the training

Table 1: Coupling functions.

(a) Standard Coupling

DoF	$f(q_1, q_2, q_3, q_4)$
q_5	$q_3 - \cos(q_2)$
q_6	$q_1 + 0.1\sin(q_2)$
q_7	$q_4\cos(q_2)$
q_8	$q_1 + q_3^2$
q_9	$1.5\sin(q_2)$
q_{10}	$-q_4 q_1$
q_{11}	$\sin(q_1)$
q_{12}	$0.4 q_3 q_4$
q_{13}	$-0.9q_1 - q_2 + q_3 - 2q_4^2$
q_{14}	$-3\sin(q_3)$
q_{15}	$-2q_3^2$
q_{16}	$-0.9q_1^2$

(b) Context-dependent Coupling

DoF	$f(q_1, q_2, q_3, q_4, l_1, l_2, l_3, l_4)$
q_5	$q_3 - \cos(2l_2 q_2)$
q_6	$q_1 + 2l_1 0.1\sin(2l_2 q_2)$
q_7	$q_4\cos(2l_4 q_2)$
q_8	$q_1 + q_3^{2\cdot 2l_3}$
q_9	$2l_2 1.5\sin(q_2)$
q_{10}	$-(l_4 + l_1)q_4 q_1$
q_{11}	$\sin(2l_1 q_1)$
q_{12}	$2l_3 0.4 q_3 q_4$
q_{13}	$-2l_1 0.9q_1 - q_2 + q_3 - 2q_4^{2\cdot 2l_4}$
q_{14}	$-2l_3 3\sin(q_3)$
q_{15}	$-2q_3^{2\cdot 2l_3}$
q_{16}	$-0.9q_1^{2\cdot 2l_1}$

dataset, we collect 100 trajectories. For each sampled link length configuration, the initial configuration of the first four joints is uniformly sampled from $[0, 30]^\circ$. For the test dataset, we sample 256 distinct link length configurations by taking all possibilities from the set $\{0.35, 0.45, 0.55, 0.65\}^4$m. For these test configurations, the initial angle for all joints is consistently set to 15°. This regular, systematic sampling for the test set is chosen to provide a comprehensive and unbiased evaluation of our model's performance across a representative grid of contextual parameters, ensuring robust assessment of context-dependent behavior.

C.4. Architecture and training details

For the 3 architectures (cAE, IcAE and McAE), we used four biorthogonal layer pairs within the encoder $\rho^{(l)} : \mathcal{R}^{n_l} \to \mathcal{R}^{n_{l-1}}$ and the decoder $\varphi^{(l)} : \mathcal{R}^{n_{l-1}} \to \mathcal{R}^{n_l}$ where the size $n_l = [8, 16, 16, 16]$. The latent size $d = n_0$ is set to 4 as the first four joints describe the entire pendulum. For the cAE, the α parameter is set to $\frac{\pi}{10}$. For the IcAE and McAE, the neuromodulation neural network is composed of 2 fully connected layers of size 4 with a hyperbolic tangent activation function after the first layer. We use the Riemannian Adam optimizer from `geoopt` with a learning rate of $5 \cdot 10^{-2}$ and a weight decay of 10^{-6} for 5000 epochs of training. We also use the *ReduceLROnPlateau* scheduler from `PyTorch` python package with a patience of 100 and a factor of 0.9. Finally, the loss used for optimizing the parameters of the different autoencoders is as follows

$$\mathcal{L}_{ae} = \frac{1}{N}\sum_{i=0}^{N} (\boldsymbol{x} - P(\boldsymbol{x}))^2 + (\dot{\boldsymbol{x}} - \nabla_{\boldsymbol{x}} P(\boldsymbol{x}) \cdot \dot{\boldsymbol{x}})^2$$

where N is the batch size, and $P(\boldsymbol{x}) = \varphi \circ \rho\ (\boldsymbol{x})$.